Microelectronics Journal 45 (2014) 530–538

Contents lists available at ScienceDirect

Microelectronics Journal

journal homepage: www.elsevier.com/locate/mejo

# Row-based body-bias assignment for dynamic thermal clock-skew compensation

Valerio Tenace, Sandeep Miryala, Andrea Calimera, Alberto Macii *, Enrico Macii, Massimo Poncino

*Dipartimento di Automatica e Informatica, Politecnico di Torino, 10129 Torino, Italy*

ARTICLE INFO

*Article history:*
Received 7 December 2012
Received in revised form
24 October 2013
Accepted 27 November 2013
Available online 4 January 2014

*Keywords:*
Clock tree
Clock skew
Thermal effects
Body biasing
Adaptive strategy
Skew compensation

ABSTRACT

This work describes how Adaptive Body Biasing can be applied on the Clock Distribution Networks (CDNs) to dynamically compensate the thermally induced skews. By selectively changing the bulk polarizations of the clock buffers, i.e., Forward Body Bias (FBB) to speed-up, or Reverse Body Bias (RBB) to slow down, it is possible to recover the timing phase shifts that accumulate along the clock tree paths due to on-chip thermal gradients.

The design constraints that are imposed by the semi-custom layout rules make the physical implementation of the proposed technique a non-trivial task. In fact, such rules require that the biasing should be applied with a relatively coarse granularity, e.g., a row of the layout rather than a single cell.

In this paper we propose a row-based Integer Linear Programming (ILP) formulation for identifying a physically constrained optimal body-biasing assignment for thermal clock skew compensation. We also describe a faster linear heuristic that accounts the timing penalties imposed on the logic circuits by the application of the row-based body-biasing.

In order to assess the applicability and the effectiveness of the proposed strategies, we benchmarked both the ILP and the heuristic against a cell-based approach using as test cases a set of circuits mapped onto an industrial 40 nm technology provided by STMicroelectronics. Experimental results show how a sensible reduction of the thermally induced skew can be achieved, while maintaining compliance with the imposed physical design rules and while keeping timing penalties under control.



## 1. Introduction

Designing the Clock Distribution Network (CDN) is one of the critical design phases in modern digital integrated circuits [1]. Firstly because the clock signal, connected to all the sequential elements, is typically loaded with very high fanout; secondly, and this is the most critical issue, because such sequential elements are irregularly scattered across the whole layout making the clock signal to spread over the entire die with a total length in the order of millimeters. Hence, guaranteeing fast propagation of clock waveforms with clean and sharp edges represents a major challenge.

The main design constraint for a reliable CDN is the *clock skew*, commonly defined as the maximum difference in the arrival time of the clock edges between any two sequential elements belonging to the same clock domain. Forcing such a constraint is mandatory to avoid race conditions between paths and reduce the risk of incorrect data latching or, even worse, circuit de-synchronization.

While regular CDN structures like buffered *trees* and *H-trees* [1–3] have historically managed to solve the clock skew issue, with the aggressive scaling of the CMOS technology, new physical effects have raided up making standard CDN design approaches inadequate. Circuits mapped with technologies below the 65 nm node show unpredictable behaviors due to parametric variations [4], and, especially when considering wire-dominated structures as the CDNs, due to temperature-induced parametric variations [5,6]. That's true not just for the higher peak temperatures devices must tolerate, but mostly because of the large spatial and temporal gradients that appear across the layout (more than 50 °C in high performance ICs as reported in [4]).

Since devices that work at different temperatures may show significant performance mismatches, branches of the CDN that cross different thermal regions may have unbalanced path delays, that is, branches over hot die regions get slower, while those crossing cold regions get faster. The resulting difference generates clock-skews [7,8], which may vary, dynamically, depending on the power density distribution.

To address the problem of CDN optimization for thermal skew, the authors of [9–11] proposed an innovative design paradigm (which is the target of this work) based on the use of *adaptive* devices, i.e., adjustable buffers whose delay can be tuned, at run-time, depending

* Corresponding author.
*E-mail address:* alberto.macii@polito.it (A. Macii).

0026-2692/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.mejo.2013.11.013

on the temperature distribution in order to compensate dynamic clock-skews. Three are the most popular embodiments of tunability based on (1) regulating delay via voltage scaling, (2) the use of programmable capacitive loads to increase the fanout [9], and (3) regulation of the body bias voltage. The latter strategy has appeared to be the most effective solution: most of the CMOS libraries offer technological support as it does not require any customization of the standard gates.

Adaptive Body Biasing (ABB) relies on the principle of applying a forward bias voltage (FBB) or a reverse bias voltage (RBB) to decrease or increase gate delays respectively. Thereby, skew variations can be compensated using FBB on those buffers that belong to the slow branches of the CDN, i.e., those that cross hot regions, and RBB to those buffers belonging to the fast branches of the CDN, i.e., those that cross cold regions.

Since the number of buffers in the CDN is typically pretty large, finding the optimal sequence of RBB/FBB configurations that brings back the CDN to have a bounded skew is not trivial and it requires dedicated optimization algorithms. The picture gets even more complicated when considering that, while for old technologies (i.e., above 90 nm) dedicated bulk contacts were available in each gate, i.e., different bulk polarizations could be independently applied to each buffer in the CDN, for nanometric technologies (65 nm and below), the picture radically changed: the internal gate contacts have been replaced with dedicated contact cells placed through the layout [12] and uniformly spaced following a row-by-row scheme in order to get a stable bulk polarization. Addressing this issue requires new models and algorithms that include the physical constraints during the optimization of the body-biasing assignment problem. This is the key contribution of this work.

Borrowing the same adaptive clock tree architecture proposed in [13], we propose an optimization strategy that can be used to identify a physical constrained optimal body-biasing configuration for thermal clock skew compensation. More specifically, we first describe an exact ILP-based formulation; then we introduce a faster linear time heuristic that does not just consider skew optimizations, but it also takes into account the timing penalties row-based body-biasing infers on the logic circuits. The optimal selection returned from both the ILP and the heuristic can be used as reference to fill the LUT of the thermal management unit (TMU) that will drive the clock-tree tuning during in-field operations [13].

Using as test cases a set of circuits mapped onto an industrial 40 nm technology by STMicroelectronics, we benchmarked our row-based strategies (ILP and heuristic) against a cell-based approach. Experimental results show that the skew induced by uneven thermal profiles can be effectively reduced while maintaining compliance with the physical design rules and, most importantly, with reduced circuit timing penalties. To complete the analysis, we also provide a comparative analysis between ABB and FBB, that is, when only FBB is used to speed-up buffers in the hot regions and compensate the overall clock skew.

The remainder of paper is organized as follows. Section 2 describes the main effects of temperature gradients on the CDN and a brief summary of the state-of-the-art in thermal-aware clock synthesis. Section 3 introduces the row-based body-biasing architecture along with the optimization algorithms we implemented. Simulation results are finally discussed in Section 4. Section 5 closes the paper with a brief summary of the work.

## 2. Background on thermally induced clock skew

### 2.1. Thermal effects on clock skew

The clock skew can be formally defined as the maximum difference between the (source to sink) arrival times $D_i - D_j$ of any pair of nodes $(i,j)$ belonging to the set of sinks $S$ of the tree:

$$Skew = |\max\{D_i - D_j\}|, \quad \forall (i,j) \in S, \text{ with } i \neq j \tag{1}$$

The arrival time at each sink is the result of the additive delay contributions of wire segments and buffers that break up the clock-path from the root. Both wires and buffers may vary their behavior significantly depending on the operating temperature. Concerning metal wires, a larger temperature gets a linear increase of the metal resistivity, as described by

$$R(x) = R_0(1 + \beta \cdot T(x)) \tag{2}$$

with $R_0$ as the resistance at the reference temperature (room temperature), $\beta$ the temperature coefficient that depends on the type of material ($3.9 \times 10^{-3}$ for Copper), and $T$ as the local temperature of the wire resulting from the diffusion of heat from the die substrate and the self-heating effect [14]. It is important to highlight that, for long wires, as the case of clock trees, $T$ may change depending on the actual position along the wire $x$ (for every 10° rise in temperature the resistance of the copper wire increases by 3.9%).

Concerning MOS transistors, several parameters show a direct dependence from temperature, e.g., threshold voltage and saturation velocity, however, carrier mobility ($\mu$) is the one that is mostly affected by temperature fluctuations.[1] Higher temperatures induce more lattice vibrations, which, in turn, cause more scattering effects among free electrons reducing the mean free path. This dependence is well modeled by

$$\mu(T) = \mu(T_0) \cdot \left(\frac{T_0}{T}\right)^m \tag{3}$$

where $T$ is the actual junction temperature and $T_0$ is the nominal temperature (about 300 K); $m$ is the temperature coefficient, which is about 1.5, but may vary depending on the process.

Eq. (4), which represents the alpha power law model [16], shows that a lower mobility brings to a smaller current capability of the active transistor, thus affecting the speed of the CMOS buffers that get slower as temperature increases:

$$I_d \propto \mu(T)(V_{dd} - V_{th})^{\alpha} \tag{4}$$

Since the clock network spans the entire die, non-uniform temperature distributions contribute negatively to the clock skew. In fact, even if the clock signal propagates on a symmetric network with both wires and buffers showing a monotonic temperature relationship, the presence of thermal gradients may cause substantial timing skew. Branches of the tree crossing hot regions get slower, while branches that run over cold regions get faster.

Fig. 1 depicts a scenario where two paths, *Source-to-Sink1* and *Source-to-Sink2*, having the same length, thus zero clock skew, can result in different delays due to the thermal effects.

### 2.2. Design strategies for skew compensation

Historically, clock network design techniques focused on the generation of minimum wire-length clock trees with zero or bounded skew [17,18], and possibly combined with wire sizing and/or with buffer insertion [19]. Such techniques assume a constant temperature along the clock network.

In [20] the authors firstly address the problem of optimizing the clock tree under spatial thermal gradients and propose to modify the traditional Deferred-Merge Embedding (DME) method proposed in [21] to search for nodes in the proximity of merging points that can minimize clock skew for both uniform and non-uniform thermal profiles.

[1] In this work we assume that CMOS devices do not show Inverted Temperature Dependence [15].